AMENDMENTS TO THE CLAIMS:

In the Claims:

1. (Currently Amended): An apparatus comprising:

a rendering engine that defines a rectangular area of pixels that bounds a triangular area of the pixels, wherein the rectangular area of pixels includes one or more lines of pixels;

the rendering engine further selects each of the one or more ~~a line~~ lines of pixels within the rectangular area of pixels, sequentially evaluates coordinates associated with the pixels of each ~~the~~ line of pixels to determine whether the pixels fall within the triangle area, ~~and~~ ceases evaluation of the coordinates associated with the pixels of each ~~the~~ line of pixels upon determining that at least one pixel of the line falls within the triangle area and a current pixel no longer falls within the triangle area, and stores information indicating which of the pixels fall within the triangle area.

2. (Original): The apparatus of claim 1, wherein the rendering engine evaluates the coordinates of the pixels in accordance with a set of linear equations that describe edges of the triangular area.

3. (Original): The apparatus of claim 2, wherein the rendering engine computes a coefficient matrix M_C for computing linear coefficients for the set of linear equations, and applies the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.

4. (Original): The apparatus of claim 3, wherein the rendering engine applies the coefficient matrix M_C to a current one of the pixels (X_C, Y_C) within the rectangular area to determine whether:

$$M_C \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_C equals:

$$M_c = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1y_2 - x_2y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2y_0 - x_0y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0y_1 - x_1y_0 \end{bmatrix} \text{and}$$

vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ are vertices of the triangular area.

5. (Original): The apparatus of claim 1, wherein the rendering engine selectively renders the pixels that fall within the triangular area by computing updated pixel data for those pixels in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

6. (Original): The apparatus of claim 5, wherein the attribute values comprise at least one of color values and texture values.

7. (Original): The apparatus of claim 5, wherein the rendering engine computes a coefficient matrix M^1 for computing linear coefficients A, B, C of the set of linear equations, and applies the coefficients A, B, C to each pixel that falls within the triangular area to compute an attribute value for the respective pixel.

8. (Original): The apparatus of claim 7, wherein the rendering engine applies the coefficient matrix M^1 to compute the linear coefficients A, B, C, for an attribute associated with vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

where the coefficient matrix M^1 equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

9. (Original): The apparatus of claim 1, further comprising a z-buffer storing a set of z-values associated with the pixels, and wherein the rendering engine compares a z-value, z_c, of the current pixel with a corresponding z-value, z_b, of a z-buffer to determine whether each pixel within the rectangular area is visible and selectively renders each pixel of the rectangular area that is visible and that falls within the triangle area.

10. (Original): The apparatus of claim 1, further comprising a control unit that issues a command to the rendering engine that specifies vertices of the triangular area.

11. (Original): The apparatus of claim 1, wherein the rendering engine comprises:

a vertex buffer for buffering the vertices of the triangular area to be rendered;

a bounding box generator that processes the vertices to compute bounding data that define the dimensions of the rectangular area; and

a rasterizer that processes the bounding data and evaluates coordinates associated with the pixel values of the rectangular area to selectively render the pixels that fall within the triangular area.

12. (Original): The apparatus of claim 11, further comprising:

an edge coefficient generator that receives the vertices buffered by the vertex buffer and processes the vertices to compute linear coefficients for a set of linear equations that describe edges of the triangular area, and

an attribute coefficient generator that processes the vertices to compute linear coefficients for a set of linear equations that describe one or more attributes associated with the triangular area, wherein

the rasterizer processes the bounding data and the coefficients in accordance with the sets of linear equations to render the pixels that fall within the triangular area.

13. (Original): The apparatus of claim 1, wherein the apparatus comprises a wireless communication device.

14. (Original): The apparatus of claim 1, wherein the apparatus comprises an integrated circuit.

15. (Original): The apparatus of claim 1, further comprising a cache memory to store at least a portion of the pixels, wherein the cache memory has a block size, and the rendering engine defines the rectangular area as a function of the block size of the cache.

16. to 31. (Cancelled)

32. (New): An apparatus comprising:

rendering means that defines a rectangular area of pixels that bounds a triangular area of the pixels, wherein the rectangular area of pixels includes one or more lines of pixels;

the rendering means further selects each of the one or more lines of pixels within the rectangular area of pixels, sequentially evaluates coordinates associated with the pixels of each line of pixels to determine whether the pixels fall within the triangle area, ceases evaluation of the coordinates associated with the pixels of each line of pixels upon determining that at least one pixel of the line falls within the triangle area and a current pixel no longer falls within the triangle area, and stores information indicating which of the pixels fall within the triangle area.

33. (New): The apparatus of claim 32, wherein the rendering means evaluates the coordinates of the pixels in accordance with a set of linear equations that describe edges of the triangular area.

34. (New): The apparatus of claim 33, wherein the rendering means computes a coefficient matrix M_C for computing linear coefficients for the set of linear equations, and applies the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.

35. (New): The apparatus of claim 34, wherein the rendering means applies the coefficient matrix M_C to a current one of the pixels (X_C, Y_C) within the rectangular area to determine whether:

$$M_C \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_C equals:

$$M_c = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1y_2 - x_2y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2y_0 - x_0y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0y_1 - x_1y_0 \end{bmatrix} \text{and}$$

vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ are vertices of the triangular area.

36. (New): The apparatus of claim 32, wherein the rendering means selectively renders the pixels that fall within the triangular area by computing updated pixel data for those pixels in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

37. (New): The apparatus of claim 36, wherein the attribute values comprise at least one of color values and texture values.

38. (New): The apparatus of claim 36, wherein the rendering means computes a coefficient matrix M^{-1} for computing linear coefficients A, B, C of the set of linear equations, and applies the coefficients A, B, C to each pixel that falls within the triangular area to compute an attribute value for the respective pixel.

39. (New): The apparatus of claim 38, wherein the rendering means applies the coefficient matrix M^{-1} to compute the linear coefficients A, B, C, for an attribute associated with vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

where the coefficient matrix M^{-1} equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{ and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

40. (New): The apparatus of claim 32, further comprising a means storing a set of z-values associated with the pixels, and wherein the rendering means compares a z-value, z_c, of the current pixel with a corresponding z-value, z_b, of a z-buffer to determine whether each pixel within the rectangular area is visible and selectively renders each pixel of the rectangular area that is visible and that falls within the triangle area.

41. (New): The apparatus of claim 32, further comprising a control means that issues a command to the rendering means that specifies vertices of the triangular area.

42. (New): The apparatus of claim 32, wherein the rendering means comprises:

means for buffering the vertices of the triangular area to be rendered;

means that processes the vertices to compute bounding data that define the dimensions of the rectangular area; and

means that processes the bounding data and evaluates coordinates associated with the pixel values of the rectangular area to selectively render the pixels that fall within the triangular area.

43. (New): The apparatus of claim 42, further comprising:

means that receives the vertices buffered by the vertex buffer and processes the vertices to compute linear coefficients for a set of linear equations that describe edges of the triangular area, and

means that processes the vertices to compute linear coefficients for a set of linear equations that describe one or more attributes associated with the triangular area, wherein

the means that processes the bounding data processes the bouding data and the coefficients in accordance with the sets of linear equations to render the pixels that fall within the triangular area.

44. (New): The apparatus of claim 32, wherein the apparatus comprises a wireless communication device.

45. (New): The apparatus of claim 32, wherein the apparatus comprises an integrated circuit.

46. (New): The apparatus of claim 32, further comprising a cache memory to store at least a portion of the pixels, wherein the cache memory has a block size, and the rendering means defines the rectangular area as a function of the block size of the cache.

47. (New): A method comprising the steps of:

defining a rectangular area of pixels that bounds a triangular area of the pixels, wherein the rectangular area of pixels includes one or more lines of pixels;

selecting each of the one or more lines of pixels within the rectangular area of pixels;

sequentially evaluating coordinates associated with the pixels of each line of pixels to determine whether the pixels fall within the triangle area,

ceasing evaluation of the coordinates associated with the pixels of each line of pixels upon determining that at least one pixel of the line falls within the triangle area and a current pixel no longer falls within the triangle area, and

storing information indicating which of the pixels fall within the triangle area.

48. (New): The method of claim 47, wherein the coordinates of the pixels are evaluated in accordance with a set of linear equations that describe edges of the triangular area.

49. (New): The method of claim 48, further including the step of computing a coefficient matrix M_C for computing linear coefficients for the set of linear equations, and applying the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.

50. (New): The method of claim 49, further including the step of applying the coefficient matrix M_C to a current one of the pixels (X_C, Y_C) within the rectangular area to determine whether:

$$M_C \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_C equals:

$$M_c = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2 y_0 - x_0 y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0 y_1 - x_1 y_0 \end{bmatrix} \text{and}$$

vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ are vertices of the triangular area.

51. (New): The method of claim 47, further including the step of selectively rendering the pixels that fall within the triangular area by computing updated pixel data for those pixels in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

52. (New): The method of claim 51, wherein the attribute values comprise at least one of color values and texture values.

53. (New): The method of claim 51, further including the step of computing a coefficient matrix M^1 for computing linear coefficients A, B, C of the set of linear

equations, and applying the coefficients A, B, C to each pixel that falls within the triangular area to compute an attribute value for the respective pixel.

54. (New): The method of claim 53, further including the step of applying the coefficient matrix M^{-1} to compute the linear coefficients A, B, C, for an attribute associated with vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

where the coefficient matrix M^{-1} equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{ and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

55. (New): The method of claim 47, further including the steps of storing in a z-buffer a set of z-values associated with the pixels, and comparing a z-value, z_c, of the current pixel with a corresponding z-value, z_b, of a z-buffer to determine whether each pixel within the rectangular area is visible and selectively renders each pixel of the rectangular area that is visible and that falls within the triangle area.

56. (New): The method of claim 47, further including the step of issuing a command that specifies vertices of the triangular area.

57. (New): The method of claim 47, further including the steps of:

buffering the vertices of the triangular area to be rendered;

processing the vertices to compute bounding data that define the dimensions of the rectangular area; and

processing the bounding data and evaluating coordinates associated with the pixel values of the rectangular area to selectively render the pixels that fall within the triangular area.

58. (New): The method of claim 57, further including the steps of:

receiving the vertices buffered by the vertex buffer and processing the vertices to compute linear coefficients for a set of linear equations that describe edges of the triangular area, and

processing the vertices to compute linear coefficients for a set of linear equations that describe one or more attributes associated with the triangular area, wherein

the bounding data and the coefficients are processed in accordance with the sets of linear equations to render the pixels that fall within the triangular area.

59. (New): The method of claim 47, wherein the method is performed in a wireless communication device.

60. (New): The method of claim 47, wherein the method is performed by an integrated circuit.

61. (New): The method of claim 47, further including the steps of storing to a cache memory at least a portion of the pixels, wherein the cache memory has a block size, and defining the rectangular area as a function of the block size of the cache.

62. (New): A computer program product stored on a computer-readable medium comprising:

code for causing a computer to define a rectangular area of pixels that bounds a triangular area of the pixels, wherein the rectangular area of pixels includes one or more lines of pixels;

code for causing a computer to select each of the one or more lines of pixels within the rectangular area of pixels;

code for causing a computer to sequentially evaluate coordinates associated with the pixels of each line of pixels to determine whether the pixels fall within the triangle area,

code for causing a computer to cease evaluation of the coordinates associated with the pixels of each line of pixels upon determining that at least one pixel of the line falls within the triangle area and a current pixel no longer falls within the triangle area, and

code for causing a computer to store information indicating which of the pixels fall within the triangle area.

63. (New): The computer program product of claim 62, wherein the coordinates of the pixels are evaluated in accordance with a set of linear equations that describe edges of the triangular area.

64. (New): The computer program product of claim 63, further including code for causing a computer to compute a coefficient matrix M_C for computing linear coefficients for the set of linear equations, and code for causing a computer to apply the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.

65. (New): The computer program product of claim 64, further including code for causing a computer to apply the coefficient matrix M_C to a current one of the pixels (X_C, Y_C) within the rectangular area to determine whether:

$$M_C \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_C equals:

$$M_c = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2 y_0 - x_0 y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0 y_1 - x_1 y_0 \end{bmatrix} \text{and}$$

vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ are vertices of the triangular area.

66. (New): The computer program product of claim 62, further including code for causing a computer to selectively render the pixels that fall within the

triangular area by computing updated pixel data for those pixels in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

67. (New): The computer program product of claim 66, wherein the attribute values comprise at least one of color values and texture values.

68. (New): The computer program product of claim 66, further including code for causing a computer to compute a coefficient matrix M^{1} for computing linear coefficients A, B, C of the set of linear equations, and code for causing a computer to apply the coefficients A, B, C to each pixel that falls within the triangular area to compute an attribute value for the respective pixel.

69. (New): The computer program product of claim 68, further including code for causing a computer to apply the coefficient matrix M^{1} to compute the linear coefficients A, B, C, for an attribute associated with vertices $v_0(x_0,y_0)$, $v_1(x_1,y_1)$, and $v_2(x_2,y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

where the coefficient matrix M^{1} equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{ and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

70. (New): The computer program product of claim 62, further including code for causing a computer to store in a z-buffer a set of z-values associated with the pixels, and code for causing a computer to compare a z-value, z_c, of the current pixel with a corresponding z-value, z_b, of a z-buffer to determine whether each pixel within

the rectangular area is visible and selectively renders each pixel of the rectangular area that is visible and that falls within the triangle area.

71. (New): The computer program product of claim 62, further including code for causing a computer to f issue a command that specifies vertices of the triangular area.

72. (New): The computer program product of claim 62, further including:

code for causing a computer to buffer the vertices of the triangular area to be rendered;

code for causing a computer to process the vertices to compute bounding data that define the dimensions of the rectangular area; and

code for causing a computer to process the bounding data and evaluating coordinates associated with the pixel values of the rectangular area to selectively render the pixels that fall within the triangular area.

73. (New): The computer program product of claim 62, further including:

code for causing a computer to receive the vertices buffered by the vertex buffer and process the vertices to compute linear coefficients for a set of linear equations that describe edges of the triangular area, and

code for causing a computer to process the vertices to compute linear coefficients for a set of linear equations that describe one or more attributes associated with the triangular area, wherein

the bounding data and the coefficients are processed in accordance with the sets of linear equations to render the pixels that fall within the triangular area.

74. (New): The computer program product of claim 62, wherein the computer program product is contained in a wireless communication device.

75. (New): The computer program product of claim 62, wherein the computer program product is executed by an integrated circuit.

76. (New): The computer program product of claim 62, further including code for causing a computer to store to a cache memory at least a portion of the pixels, wherein the cache memory has a block size, and code for causing a computer to define the rectangular area as a function of the block size of the cache.